



# Administration

Base de données

PostgreSQL

Directives





# Post-installation

- Post installation
  - La log de l'installation se trouve dans le répertoire
    - C:\Program Files\PostgreSQL\data\logs\pg96
  - Contenu du fichier install.log
    - Les fichiers de ce cluster appartiendront à l'utilisateur « tellora ».
    - Le processus serveur doit également lui appartenir.
    - L'instance sera initialisée avec la locale « French\_France.1252 ».
    - La configuration de la recherche plein texte a été initialisée à « french ».
    - Les sommes de contrôles des pages de données sont désactivées.
    - correction des droits sur le répertoire existant C:/Program Files/PostgreSQL/data/pg96... ok
    - création des sous-répertoires... ok
    - sélection de la valeur par défaut de max\_connections... 100
    - sélection de la valeur par défaut pour shared\_buffers... 128MB
    - sélection de l'implémentation de la mémoire partagée dynamique...windows
    - création des fichiers de configuration... ok
    - lancement du script bootstrap...ok
    - exécution de l'initialisation après bootstrap...ok
    - synchronisation des données sur disqueok



# Post-installation

- Services Windows

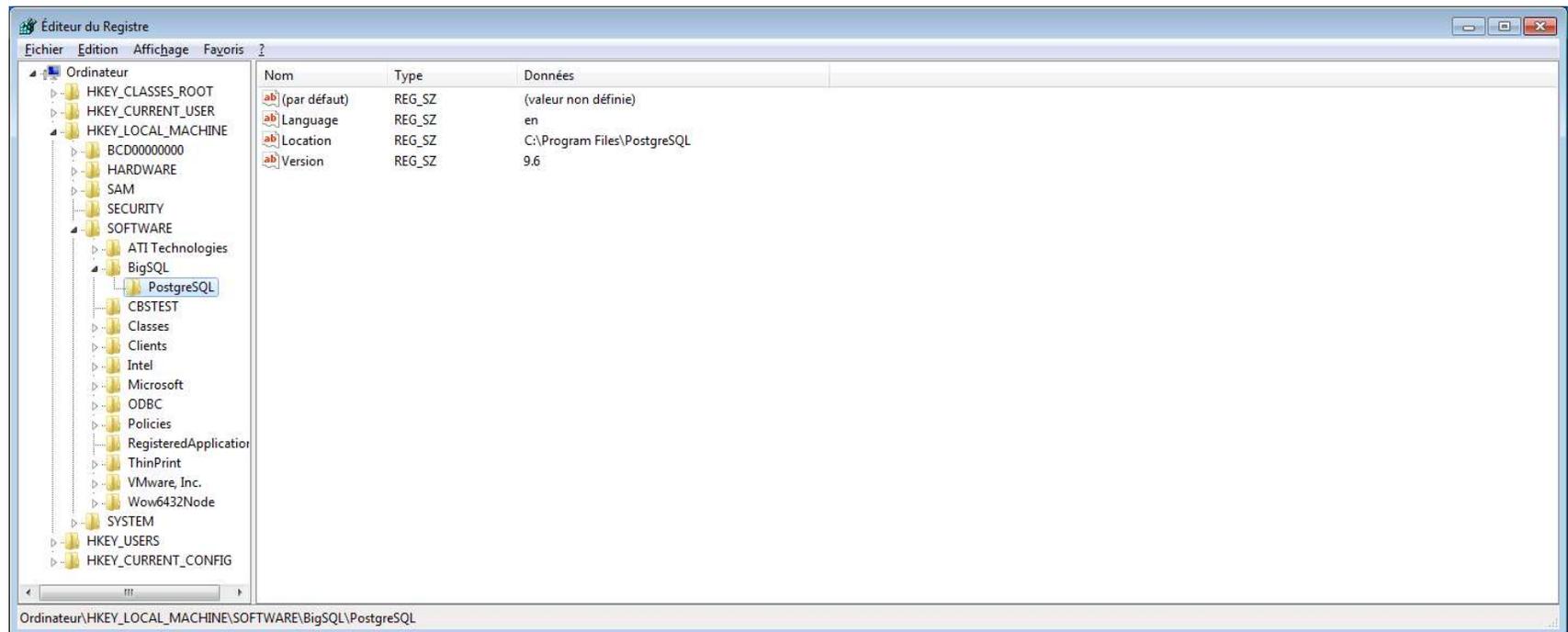
The screenshot shows the Windows 'Gestion de l'ordinateur' (Computer Management) window. The 'Services' console is open, displaying a list of system services. The 'PostgreSQL 9.6 Server' service is selected and highlighted in blue. The table below represents the data shown in the screenshot.

Nom	Description	État	Type de démarrage	Ouvrir une session en
Pare-feu Windows	Le Pare-f...	Dém...	Automatique	Service local
Parental Controls	Ce servic...		Manuel	Service local
Partage de connexion Internet (ICS)	Assure la ...	Désactivé		Système local
Planificateur de classes multimédias	Active la ...		Automatique	Système local
Planificateur de tâches	Permet à ...	Dém...	Automatique	Système local
Plug-and-Play	Permet à ...	Dém...	Automatique	Système local
<b>PostgreSQL 9.6 Server</b>	The worl...	Dém...	Automatique	Système local
Prise en charge de l'application Rapports et solu...	Ce servic...		Manuel	Système local
Programme d'installation ActiveX (AxInstSV)	Valide le ...		Manuel	Système local
Programme d'installation pour les modules Win...	Permet l'i...		Manuel	Système local
Propagation du certificat	Copie de...		Manuel	Système local
Protection logicielle	Permet le...	Dém...	Automatique (débu...	Service réseau
Protocole EAP (Extensible Authentication Protoc...	Le service...		Manuel	Système local
Protocole PNRP	Permet la...		Manuel	Service local
Publication des ressources de découverte de fon...	Publie ce...	Dém...	Manuel	Service local
Redirecteur de port du mode utilisateur des serv...	Permet la...		Manuel	Système local
Registre à distance	Permet a...		Manuel	Service local
Routage et accès distant	Offre aux ...	Désactivé		Système local
Sauvegarde Windows	Offre des ...		Manuel	Système local
Serveur	Prend en ...	Dém...	Automatique	Système local
Serveur de priorités des threads	Permet l'...		Manuel	Service local
Service de biométrie Windows	Le service...		Manuel	Système local
Service de cache de police Windows	Optimise ...	Dém...	Automatique (débu...	Service local
Service de chiffrement de lecteur BitLocker	BDESVC ...		Manuel	Système local
Service de configuration automatique WLAN	Le service...		Manuel	Système local
Service de configuration automatique WWAN	Ce servic...		Manuel	Service local
Service de découverte automatique de Proxy We...	WinHTTP...		Manuel	Service local
Service de l'Assistant Compatibilité des program...	Ce servic...	Dém...	Automatique	Système local



# Post-installation

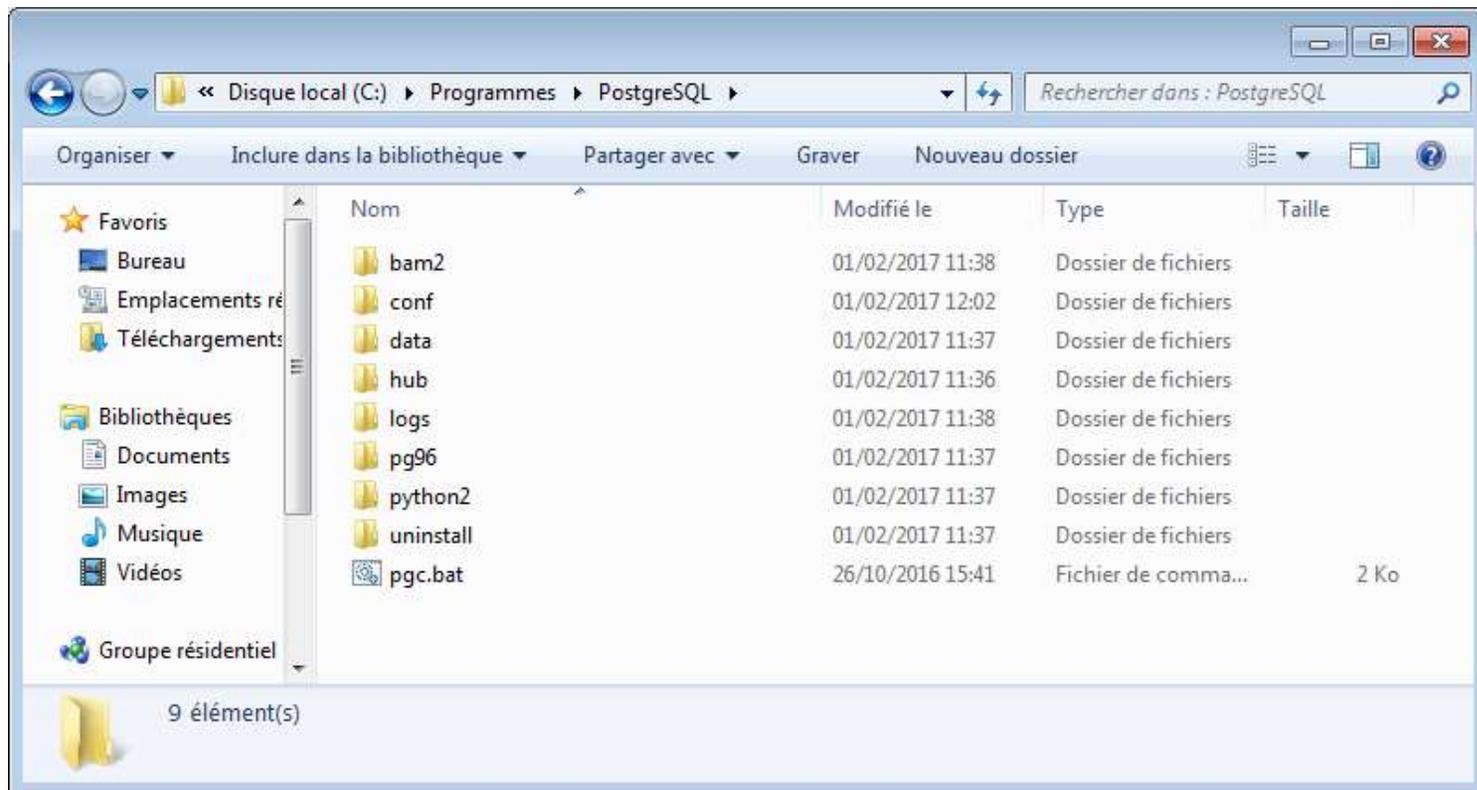
- Clés de registres





# Post-installation

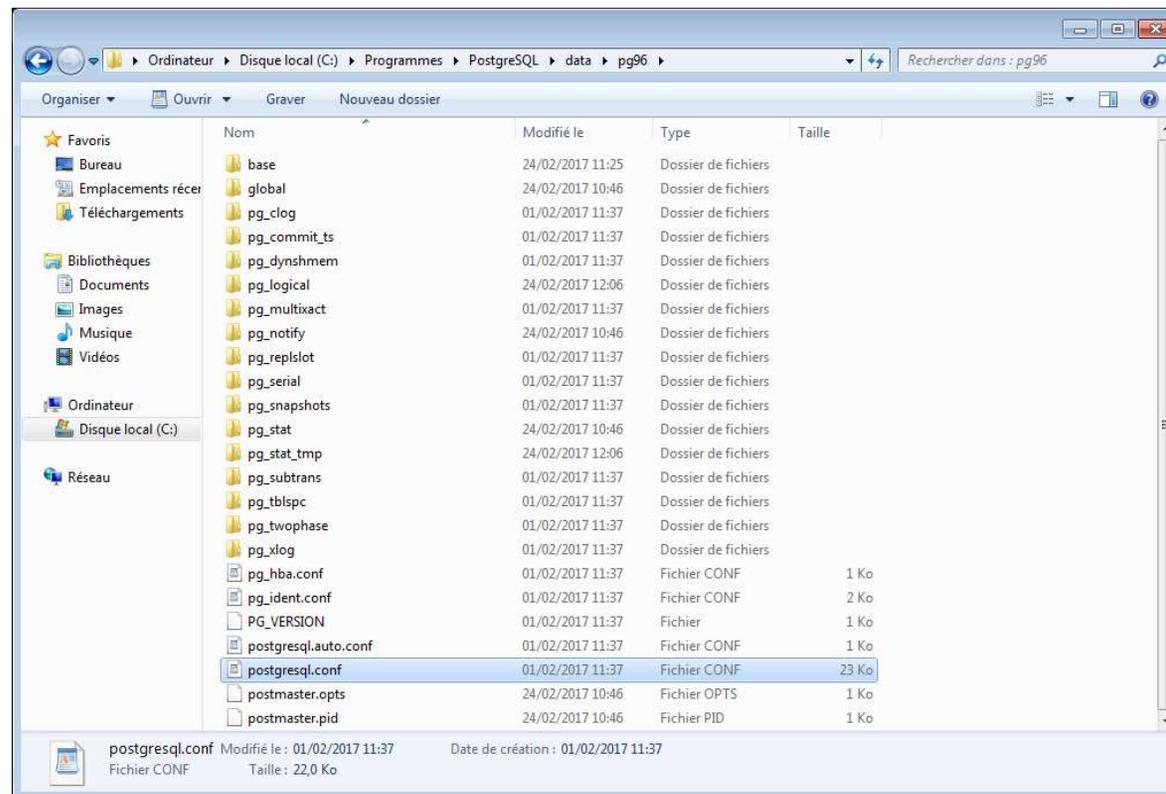
- répertoires du noyau





# Post-installation

- dans C:\Program Files\PostgreSQL\data\pg96
  - Se trouvent les répertoires du noyau PostgreSQL





# Arrêt/Démarrage du serveur

---

- Commande `pg_ctl`
  - Permet de démarrer le serveur PostgreSQL avec des options permettant d'arrêter, de redémarrer ou de recharger le serveur
    - Start = démarre le serveur
    - Stop = arrête le serveur
    - Restart = redémarre le serveur
    - Reload = recharge le serveur
  - Options
    - -D = précise le répertoire des données et fichiers de configuration
    - -l = redirige les messages vers un fichier de traces



# Arrêt/Démarrage du serveur

---

- Commande `pg_ctl`, sous windows
  - L'option « register », précise le nom du service Windows
    - -N = nom du service
    - -P = mot de passe du user système
    - -D = emplacement du groupe de base de données
    - Exemple
      - `Pg_ctl register -N pgsql9.6.1-1 -U postgres -P secret -D c:\PostgreSQL\data`
      - Enregistrer une nouvelle instance de PostgreSQL, attachée au service Windows nommé « `pgsql9.6.1-1` », le user `postgres` est administrateur de cette instance et a le mot de passe « `secret` », le répertoire des données est `c:\PostgreSQL\data`



# Arrêt/Démarrage du serveur

---

- Commande pg\_ctl, sous windows (suite)
  - Retirer un service Windows et désactiver l'instance PostgreSQL
    - Pg\_ctl unregister -N pgsql9.6.1-1
    - Cette commande désactive l'instance dans le panneau de configuration
      - Le service Windows disparaîtra après redémarrage du système



# Arrêt/Démarrage du serveur

- Scripts sous Linux
  - Les version Debian (version 4) et Ubuntu disposent de scripts supplémentaires
    - Le paquetage = postgresql-common contient les scripts
      - /usr/bin/pg\_lsclusters = liste les instances avec leur configuration et l'emplacement des données
      - Exemple
      - [root]# pg\_lsclusters
      - | Version | Cluster  | Port | Status | Owner    | Datadirectory        | Logfile                                      |
|---------|----------|------|--------|----------|----------------------|--|
| 9.6     | pgsql961 | 5434 | online | postgres | /var/lib/postgresql/ | pgsql961<br>/var/lib/postgresql/pgsql961.log |
      - » La première colonne affiche la version, puis le nom de l'instance, le port, le statut puis le nom du super administrateur, le répertoire dans lequel sont stockées les données puis le nom du fichier de trace



# Arrêt/Démarrage du serveur

- Scripts sous linux (suite)
  - Le paquage = postgresql-common (suite)
    - /usr/bin/pg\_createcluster = crée une instance, il utilise « initdb »
    - /usr/bin/pg\_ctlcluster = arrête ou redémarre les instances en utilisant la commande pg\_ctl avec 3 arguments
      - » La version
      - » Le nom de l'instance
      - » L'action à effectuer
      - » Exemple :
        - Pg\_ctlcluster 9.6 pgsq1961 stop
    - /usr/bin/pg\_upgradecluster = upgrade une instance
    - /usr/bin/pg\_dropcluster = supprime une instance
    - /usr/sbin/pg\_maintenance = effectue des taches de maintenance sur les instances
  - postgresql-client-common = utilise un script pg\_wrapper qui permet d'utiliser une commande client en fonction de l'instance sur laquelle le client veut se connecter. Ce script permet d'utiliser la bonne version d'une commande en fonction de l'instance



# Outil PSQL

- L'outil PSQL est un outil de commandes permettant de saisir des commandes SQL
- Lancer l'outil en tapant :
  - PSQL

```
PSQL
psql (9.6.1)
Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
           Les caractères 8 bits peuvent ne pas fonctionner correctement.
           Voir la section « Notes aux utilisateurs de Windows » de la page
           référence de psql pour les détails.
Saisissez « help » pour l'aide.
postgres=# _
```



# Outil PSQL

- Aide en ligne de l'outil PSQL
  - postgres=# help
    - Vous utilisez psql, l'interface en ligne de commande de PostgreSQL
    - Saisissez:
      - \copyright pour les termes de distribution
      - \h pour l'aide-mémoire des commandes SQL
      - \? pour l'aide-mémoire des commandes psql
      - \g ou point-virgule en fin d'instruction pour exécuter la requête
      - \q pour quitter
    - postgres=#



# Outil PSQL

- Quelques commandes utiles
  - Psql -l, liste des bases de données
  - \db(+), liste des tablespaces
  - dg(S+), liste des rôles
  - \d[S+] , liste des tables, vues et séquences
  - \dl , ou \lo\_list, affiche la liste des «Larges Objects »
  - \H , bascule le mode de sortie HTML
  - Dans l'outil psql
    - select \* from table ; => voir son contenu d'une table
    - \d table => voir la structure de ta table (ou de la vue)
  - Système
    - \cd [REPERTOIRE], change de répertoire de travail
    - \setenv NOM [VALEUR] (dés), initialise une variable d'environnement
    - \! [COMMANDE] , exécute une commande shell



# Outil PSQL

- Quelques commandes utiles d'affichage
  - `\pset [NOM [VALEUR]]` règle l'affichage de la table
    - (NOM :=  
{format | border | expanded | fieldsep | fieldsep\_zero | footer | null |
    - numericlocale | recordsep | recordsep\_zero | tuples\_only | title | tableattr | pager |
    - unicode\_border\_linestyle | unicode\_column\_linestyle | unicode\_header\_linestyle})



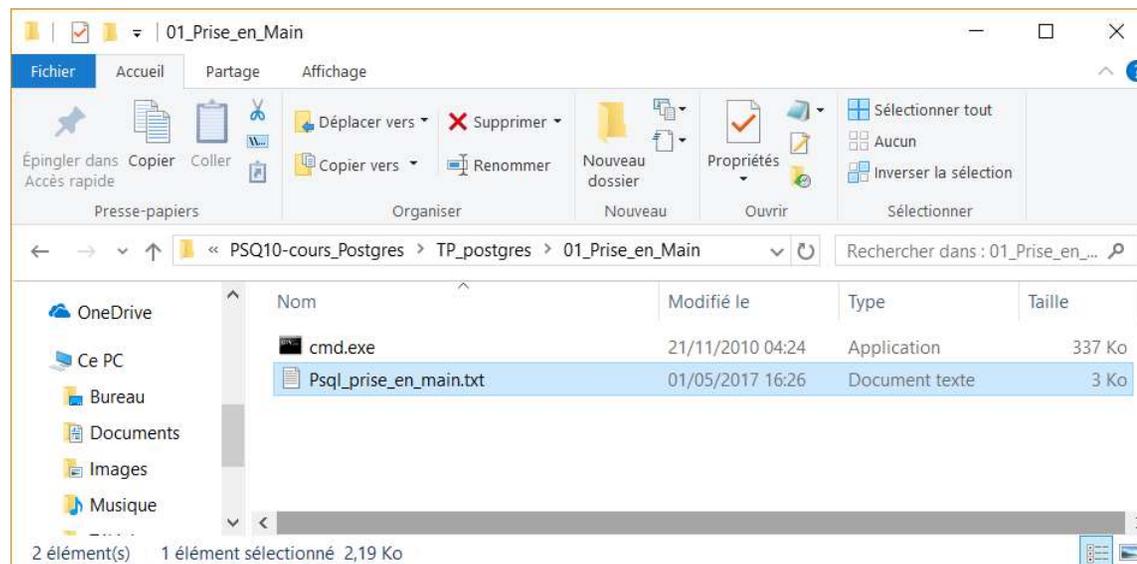
# Outil PSQL

- Commandes de connexion
  - `\c[onnect] {[NOM_BASE]- UTILISATEUR]- HOTE]- PORT]- ]  
| conninfo}`
    - se connecte à une autre base de données
      - (actuellement connecté à postgres )
    - `\encoding [ENCODAGE]` affiche ou initialise l'encodage du client
    - `\password [UTILISATEUR]`, modifie de façon sécurisé le mot de passe d'un utilisateur
    - `\conninfo` , affiche des informations sur la connexion en cours
  - Exemple
    - `postgres-# \conninfo`
    - Vous êtes connecté à la base de données « postgres » en tant qu'utilisateur « postgres » sur l'hôte « localhost » via le port « 5432 ».



# Travaux pratiques

- Dans le répertoire TP\_Protgres\01\_prise\_en\_main
- Recopiez et Exécutez les commandes du script
  - Psql\_prise\_en\_main.txt
- Puis ouvrez le cahier de travaux pratiques et
  - Faire l'exercice 01 prise\_en\_main





# Base de données

---

- Base de données
  - Une base de données est un conteneur de schémas et indirectement de tables et d'index et autres objets de schéma
  - A la connexion d'une application dans une base de données seuls les objets de cette base de données sont visibles
    - Les objets n'appartenant pas à cette base de données ne sont pas visibles
    - Il est important de répartir correctement les applications dans les bases de données en utilisant la notion de schéma



# Notion de Tablespace

---

- Tablespace
  - Un tablespace est un répertoire d'un système de fichiers dans lequel PostgreSQL écrit les fichiers correspondant aux tables et aux index
  - Par défaut PostgreSQL dispose d'un tablespace situé dans le répertoire du groupe de bases de données
    - DATA, précisé au moment de l'installation du serveur
    - Il est possible de créer d'autres tablespaces permettant de choisir l'emplacement du stockage d'une table ou d'un index
      - Par exemple sur un ensemble de disques dédiés et différents Pour une meilleure répartition des écritures



# Notion de Tablespace

- Création de tablespaces
  - L'utilisation de tablespaces permet à une instance de stocker les fichiers à un autre emplacement
    - `CREATE TABLESPACE nom_TS [OWNER nomrole] LOCATION 'nom_repertoire' ;`
      - Par défaut le tablespace appartient à l'utilisateur qui exécute la commande
      - Seul un super user peut créer un tablespace (mais il peut transmettre le privilège à un autre user)
  - Le tablespace sera utilisable au moment de la création (ou modification) des tables et des index
  - Il permet de regrouper physiquement les tables sur disque

# Notion de Tablespace



- Modification d'un tablespace
  - Il est possible de modifier le nom d'un tablespace ou le propriétaire du tablespace
    - ALTER TABLESPACE nom\_TS RENAME TO nouveau\_nom ;
    - ALTER TABLESPACE nom\_TS RENAME OWNER TO nouveau\_role ;

# Notion de Tablespace



- Suppression d'un tablespace
  - La suppression d'un tablespace peut se faire à condition que celui-ci soit vide
    - `DROP TABLESPACE [IF EXISTS] nom_TS ;`
      - IF EXISTS = évite l'apparition d'une erreur si celui-ci n'existe pas
  - Une fois que le tablespace est supprimé il est possible de supprimer le répertoire qui y était rattaché



# Notion de base de données

- Dans le répertoire « base » PostgreSQL crée un répertoire par nouvelle database créée
  - Le nom du répertoire correspond à un identifiant référencé et associé à la database dans le catalogue système
    - `Select oid, datname from pg_catalog.pg_database ;`
  - Dans chaque répertoire de database les fichiers qui correspondent aux tables et aux index sont nommés avec un identifiant qui lui est associé
    - `Select relfilenode, relname from pg_catalog.pg_class ;`
    - Si la taille d'un fichier dépasse 1 giga, PostgreSQL crée un nouveau fichier en utilisant le : `relfilenode + numéro séquentiel`
      - `Relfilenode.1` puis `relfinode.2`



# Notion de base de données

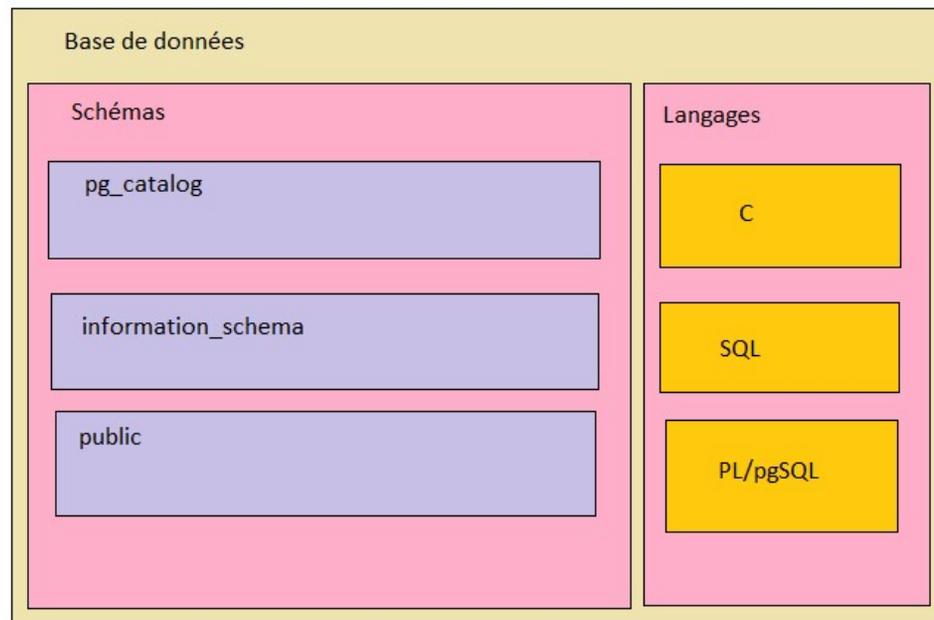
---

- Bases de données
  - A l'installation d'un serveur PostgreSQL sont créées 3 bases de données
    - Postgres
    - Template0
    - Template1
  - La création d'une base de données utilisateur se fait en copiant la base de référence « template1 »
    - Il est possible de copier une autre base de référence avec la clause
      - Template nom\_base
      - La base de référence sera entièrement copiée, structure et données
    - Il sera possible de créer un modèle de base de référence
    - Il est possible de créer une base avec un jeu de caractère différent avec la clause « encoding »



# Notion de base de données

- Une base de données nouvellement créée contient un ensemble d'objets
  - Au premier niveau on trouve les schémas et les langages





# Notion de base de données

---

- Création d'une base de données
  - CREATE DATABASE commandes  
ENCODING 'UTF8' template template1 ;
- Il est possible de supprimer une base de données et tous les objets qu'elle contient
  - DROP DATABASE commandes ;
- Dictionnaire de données
  - Select \* from pg\_catalog.pg\_database;
    - Affiche le nombre de bases de données
  - Select \* from pg\_langage;
  - Affiche les langages de programmation disponibles et accessible pour écrire des procédures



# Travaux Pratiques

- Ouvrez le cahier de travaux pratiques et
  - Faire l'exercice 02 Créer base
- Utilisez le répertoire
  - TP\_Postgres\02\_creebase,

